An Operating System Kernel for Manycore Architectures

Yutaka Ishikawa†‡, Atsushi Hori‡, Balazs Gerofi‡, and Akio Shimada‡
† University of Tokyo ‡ Riken AICS

A manycore architecture is one of the promising approaches to building exascale systems. There are several ways to realize a compute node using manycore architectures, i.e., accelerator/coprocessor, heterogeneous CPU, and manycore only. The University of Tokyo and Riken Advanced Institute of Computational Science have been designing and developing system software for manycore-based supercomputers, especially coprocessor-type and manycore only node architectures. Figure 1 depicts the operating system kernel for the target architectures that we have been being designed and developed. Linux kernel runs in the host CPU while a light-weight micro kernel runs in manycore units in case of a co-processor type manycore architecture. In case of a standalone manycore architecture, Linux kernel runs on some cores and light-weight micro kernels run on the other cores. The current target platform is an Intel MIC architecture in which compute nodes, each of which consists of manycore units with a host machine, are connected by Infiniband.

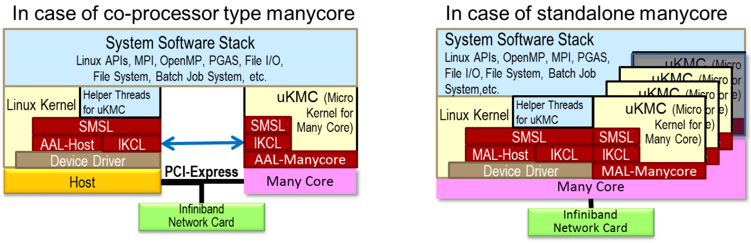

Figure 1: Software Architecture

We have been focused on the left hand-side configuration of Figure 1. AAL, Accelerator Abstraction Layer, has been designed to hide hardware-specific functions of manycore units connected with host via PCI Express. It provides the kernel programming interfaces to operating system developers. The AAL in the host is implemented as a Linux device driver. The IKCL is the inter-kernel communication layer that realizes the data transfer and signal notification between the host and manycores. The SMSL is the system service layer on top of AAL that provides low-level kernel programming interfaces to operating system developers. A lightweight micro kernel for manycore units is implemented on top of AAL, IKCL, and SMSL. Currently, just a prototype microkernel has been implemented to test functionalities of AAL, IKC, and SMSL. A Linux kernel runs in the host CPUs to perform rich OS functions such as file systems and high-level communication facilities, whose footprints are large enough to pollute memory cache of manycores if such a function is executed in manycore units. A low-level small latency communication facility, called DCFA, and file I/O offloading methods have been designed, implemented, and evaluated[5, 2].

There are mainly two approaches to provide an operating system for compute nodes, i.e., Linux-based and lightweight micro kernels. ZeptoOS[1] for IBM Blue Gene/P and CNL for Cray XT are examples of Linux-based operating systems. There are several lightweight operating system kernels for compute nodes, such as CNK for Blue Gene/P, Catamount for Cray XT, and Kitten[4]. Several operating system researches for manycores have been conducted. For example, Helios[3] provides a single interface of managing heterogeneous systems, especially systems with programmable devices, which offloads a part of kernel functions to the devices.

1 Challenges addressed

- Cache-aware system software stack: Because manycores have small memory caches and limited memory bandwidth, the footprint in the cache during both user and system program executions should be minimized. We have addressed cache-aware file I/O methods[2].

- Scalability: One of the scalability issues results from the internal data structures to manage resources shared by cores, such as thread and memory, in an operating system kernel. A light-weight microkernel being designed and implemented minimizes such shared structures to reduce contentions.
- Data movement optimization: Manycore architectures are getting a deep memory hierarchy to provide larger memory capacity than manycore's main memory. For example, main memory in manycore units and host memory are utilized to execute an application. To hide data movement between host and manycore and overlap computation and such data movement, the new paging system is now being designed and implemented.
- Minimum overhead of communication facility: Minimum overhead of communication between cores as well as direct memory access between manycore units is required for strong scaling. A prototype facility has been designed, implemented, and evaluated[5].
- Portability: The system software stack should support portability of existing programs running in PC clusters.

2 Maturity

The current lightweight micro kernel implementation is not a concrete kernel design, but rather it is for a proof of concepts. A product level implementation just starts with Japanese manufacturing companies.

3 Uniqueness and Novelty

The feature of the AAL, IKCL and SMSL is a novel feature for manycore architectures though the idea of a hardware abstraction layer itself is common approach to providing portability. This software stack does not exclude other micro kernels for manycores if those are implemented on top of this software stack. The proposed environment allows the users to invoke their specific micro kernels for applications dynamically from Linux which resides in host or part of manycores. Thus, multiple micro kernels may exist in a center operation.

4 Applicability

The current implementation is based on the Intel MIC architecture, but it would be applicable to other coprocessors and stand-alone type manycore architectures. Especially, we are interested in porting the system to K computer and FX10, commercialized version of K.

5 Effort

A Japanese two-year program, called Feasibility Study on Future HPC R&D, starts from July 2012. In the context of the program, we are going to implement a part of the proposed operating system kernel with Japanese vendors. This is not a whole operating system, and it is not intended to develop all system software by ours. We would like to have international collaboration.

References

[1] ZeptoOS, http://www.mcs.anl.gov/research/projects/zeptoos/.

[2] Yuki Matsuo, Taku Shimosawa, and Yutaka Ishikawa. A File I/O System for Many-Core Based Clusters. In *Proceeding of ROSS 2012*, 2012.

[3] Edmund B. Nightingale, Orion Hodson, Ross McIlroy, Chris Hawblitzel, and Galen Hunt. Helios: Heterogeneous Multiprocessing with Satellite Kernels. In *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles, SOSP '09*, 2009.

[4] Kevin Pedretti. Kitten: A Lightweight Operating System for Ultrascale Supercomputers. https://software.sandia.gov/trac/kitten.

[5] Min Si and Yutaka Ishikawa. Design of Direct Communication Facility for Many-Core based Accelerators. In *Proceeding of CASS 2012*, 2012.